2004 年度 修士論文

ENUM サービスを制御する新しい方法と認証

提出日： 2005 年 2 月 2 日

指導：後藤滋樹教授

早稲田大学 大学院理工学研究科 情報・ネットワーク専攻
学籍番号： 3603U158-8

李 嘉驃

目次

1 序論 **5**

1.1 背景 5

1.2 目的 5

1.3 構成 6

2 ENUM の概要 **7**

2.1 ENUM の動作 8

2.2 DDDS と NAPTR 9

2.3 ENUM のサービス 12

3 ENUM サービスを選択する新しい方法と認証方式 **13**

3.1 概要 13

3.2 通信方式 15

3.2.1 コマンドを送受信する方法 15

3.2.2 端末の識別方法 15

3.2.3 コマンドの形式 15

3.2.4 AUS 15

3.2.5 key の生成 16

3.3 認証 17

3.4 通信の過程 17

4 実験 **21**

4.1 実験の環境 21

4.2 実験の内容 23

4.2.1 実験 1 23

4.2.2 実験 2 23

4.2.3 実験 3 23

4.3 ENUM DNS サーバの設定 23
4.3.1 named.conf 23
4.3.2 ゾーンファイル (db.mydomain.com) 24
4.4 実験の結果 24
4.4.1 実験 1 の結果 24
4.4.2 実験 2 の結果 28
4.4.3 実験 3 の結果 30

5 結論 **32**
5.1 まとめ 32
5.2 今後の課題 32

謝辞 **33**

参考文献 **34**

A 用語の定義 **35**

図一覧

2.1 ENUM の概念図 7
2.2 DDDS の仕組み 10

3.1 通信の略図 14
3.2 通信の過程 18
3.3 通信の流れ 20

4.1 ネットワーク構成図 22

表一覧

2.1 ENUM のサービス 12

3.1 コマンドの形式 15
3.2 本研究用 AUS の形式 16
3.3 本研究用 AUS の問い合わせコード 16

4.1 使用した PC の仕様 21

第 1 章

序論

1.1 背景

パソコンの急速な普及を背景に、インターネットを活用した通信システムの整備が国際的に進んでいる。 2004 年第 1 四半期末 (2004 年 3 月 31 日時点) の世界のブロードバンド総回線数は 1 億 1170 万に達した。 2003 年末から 3 カ月間で約 1230 万回線の純増を記録したとされており、1 億の大台を突破している。

インターネット常時接続によるユーザの利便性向上で、個人ユーザ向けあるいは企業向けの様々なサービスが出現する。電子メール、 Web のみでなく、 Fax や電話もインターネット上で利用されるようになってきている。このような様々な通信サービスに対して、電話番号という統一的な識別子で通信相手を指定する技術が ENUM である。

現在、 IETF および ITU-T において、 ENUM の技術標準、管理運用手順が提案されている。海外諸国においては、すでに ENUM のトライアル運用が始まっており、そこでは通信アプリケーション、通信サービスの実験を通した技術的検証や諸問題の解決に向けた動きが加速している。日本においても、実際に ENUM をトライアル 運用することにより ENUM を用いた通信アプリケーション、通信サービスを実験する環境を構築し、技術的検証、課題整理等が進められている。

1.2 目的

家中の家電製品などをネットワークにつなぎ、便利に使いこなそうという「ネット家電」がいよいよ一般化する。本論文では、家の外から家庭内のネットワークマシンを制御して、 ENUM を使って、適切なサービスを選択する方法を研究とする。その際に、セキュリティを考慮して、外と家の中と通信する時に、相手を確認する認証方法を実装し、実用化する。

1.3 構成

第 1 章 序論

研究の概要について述べる。

第 2 章 ENUM の概要

ENUM の基本、特に本論文に関係する事項を述べる。

第 3 章 ENUM サービスを選択する新しい方法と認証方式

外から ENUM サービスをコントロールする方法と認証方式を述べる。

第 4 章 実験

実験の内容と結果を述べる。

第 5 章 結論

本論文の結論を述べる同時に、今後の課題を提起する。

第 2 章

ENUM の概要

ENUM は、電話番号をキーとして、 DNS を検索することにより、その番号に対応した利用できるアプリケーションのサービス情報を得る仕組みである。 ENUM の概念図を図 2.1 に示す。

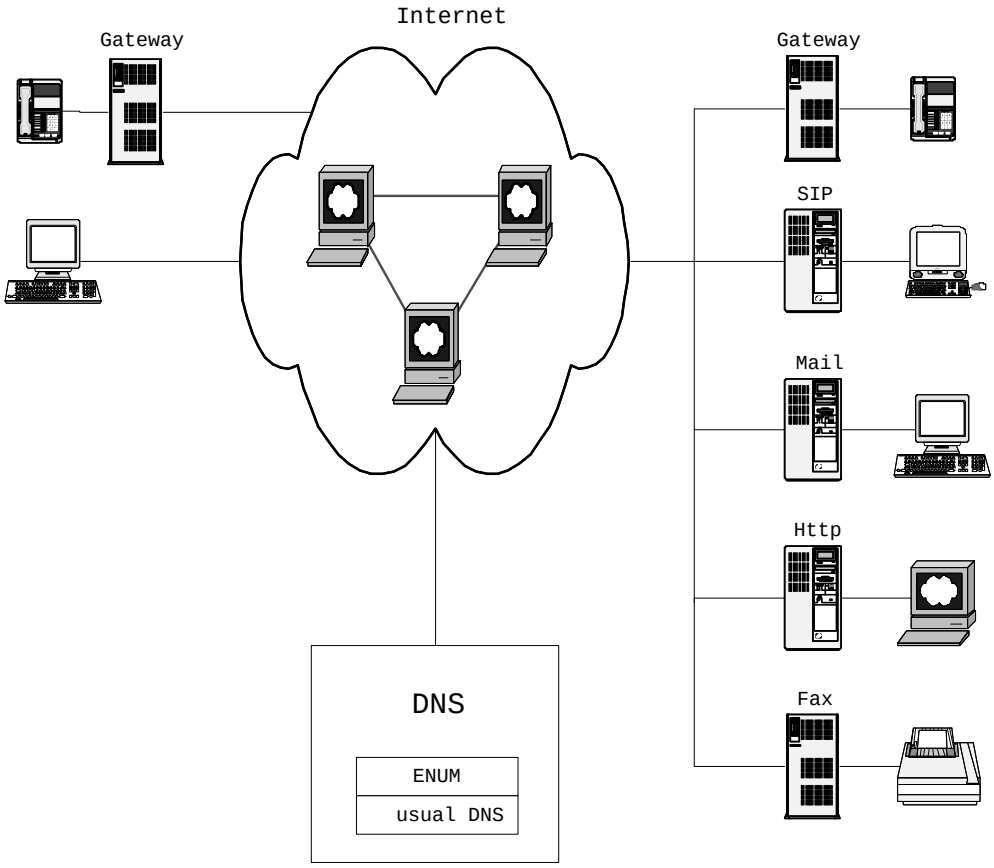

図 2.1: ENUM の概念図

2.1 ENUM の動作

例えば ”03-1234-5678” という電話番号は、 ENUM を使って以下の 6 段階の手順によりドメイン名へ変換される。

1. ITU-U によって定められた国番号付きの E.164 番号にする。

   +81-3-1234-5678

2. 先頭の ”+” と数字以外の文字を抹消する (ここで ”+” を抹消しないのは、将来 E.164 以外の別の番号体系にも対応できるようにするため)。これが後述する ENUM DDDS の検索用の文字列 AUS(Application Unique String) である。

   +81312345678

3. 数字以外の文字を抹消する。

   81312345678

4. それぞれの数字の間にドット (”.”) を挿入する。

   8.1.3.1.2.3.4.5.6.7.8

5. 数字を逆順にする。

   8.7.6.5.4.3.2.1.3.1.8

6. 最後に ENUM のドメイン ”.e164.arpa” を追加する。

   8.7.6.5.4.3.2.1.3.1.8.e164.arpa

こうして得た文字列をドメイン名として、 DNS に NAPTR リソースレコードを要求する。

このドメイン名に対して、以下のような NAPTR リソースレコードが DNS 上に登録されていた場合には、

```
$ORIGIN 8.7.6.5.4.3.2.1.3.1.8.e164.arpa
IN NAPTR 100 10 "u" "E2U+sip" "!^.*$!sip:info@sip.goto.info.waseda.ac.jp!" .
```

結果として、 URI

```
sip:info@sip.goto.info.waseda.ac.jp
```

を得る。これによりアプリケーションプログラムは、 sip:info@sip.goto.info.waseda.ac.jp に対して、 SIP を用いてセッションを確立できる。

NAPTR 行のサービスフィールド (E2U+sip) と変換規則 regexp の URI を書き換えることで、さまざまな通信サービスを、電話番号に対応付けることができる。このことは後述する。

2.2 DDDS と NAPTR

DDDS(Dynamic Delegation Discovery System) は、動的な文字列変換規則をアプリケーション内の文字列 (AUS) に適応して、 DNS を介して URI などの結果を得るシステムである。

ENUM の基本的なアルゴリズム (図 2.2) は、

1. AUS が与えられると、アプリケーションごとに規定された最初の変換により、データベースをひく鍵 (ドメイン名) をつくる。
2. 鍵をもとに DDDS データベース (DNS) をひき、変換規則を得る。
3. AUS に対して、変換規則を適用する。
4. 変換規則が最終結果を出すものでなければ、変換結果を鍵として 2 に戻る。
5. 最終結果を出す変換の結果が出力となり、 URI やドメイン名、アドレスが得られる。

変換規則をデータベースに正規表現で記述するのに用いるのは NAPTR リソースレコードである。 ENUM では、 E.164 番号から生成されるドメイン名ごとに、対応するアプリケーションを URI 形式で NAPTR リソースレコードの中に登録される。

NAPTR リソースレコードの書式を以下に示す。

```
IN NAPTR order preference flags service regexp replacement
```

- order
  16 bit 符号なし整数を使う。小さいものほど優先順位が高い。
  order は preference よりも優先する。

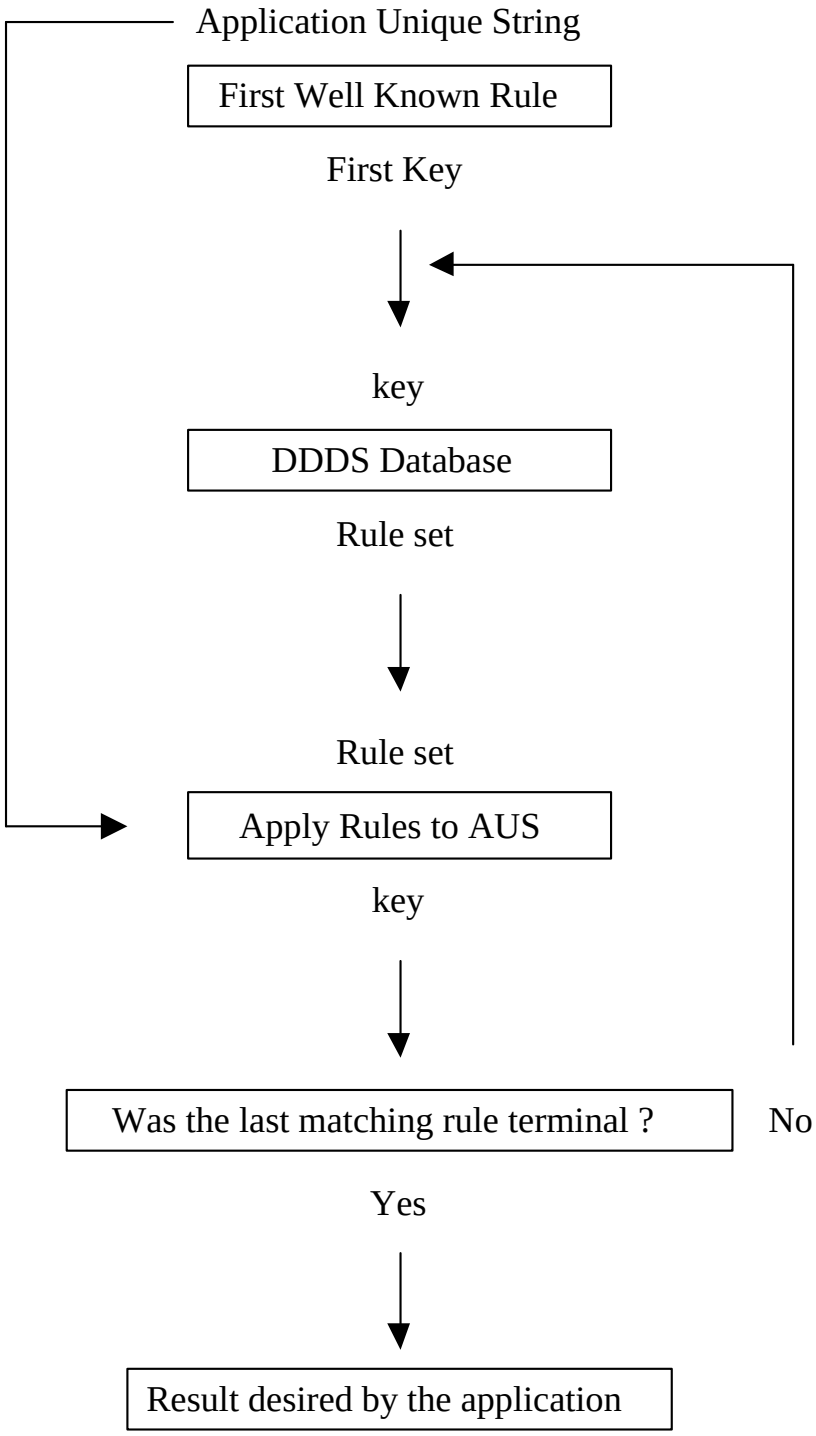

図 2.2: DDDS の仕組み

- perference
  16 bit 符号なし整数を使う。小さいものほど優先順位が高い。

- flags
  文字を使って、得られた結果が最終結果かどうかを判定する。
  ”u”は最終結果が URI であると定義されている。

- service
  使うプロトコルである E2U (ENUM to URI)+ サービス名である。

- regexp
  正規表現形式で AUS からの変換規則である。

- replacement
  固定ドメイン名を返す場合のみ、 regexp の代わりにこのフィールドにドメイン名を書く。
  それ以外の場合はドット ”.”を書く。

regexp フィールドと replacement フィールドは排他関係にある。両方に記述がある NAPTR リソースレコードは、無視されるか、エラーになる。

一回の問い合わせで複数の NAPTR リソースレコードが得られた場合、まず、 service フィールドをみて、 order が最も小さいものを選ぶ。 order が同じものが複数存在する場合は、 preference が最も小さいものを選ぶ。

同じ order 値に NAPTR リソースレコードが複数存在した場合、 preference の小さい順でリソースレコードを処理する。ある order 値の NAPTR リソースレコードが全て失敗した場合、それより大きな order 値のリソースレコードを処理せずにエラーとする。

2.3 ENUM のサービス

ENUM で使うプロトコルとして E2U (Enum to URI) が定義されている。その中で想定されているサービスのサービス・プロトコルと NAPTR の service フィールド、 URI スキームの例を表 2.1 に示す。

表 2.1: ENUM のサービス

サービス・プロトコル	service フィールド	URI スキームの例
SIP	E2U+sip	sip:info@sip.goto.info.waseda.ac.jp
H.323	E2U+h323	h323:info@goto.info.waseda.ac.jp
インターネット FAX	E2U+ifax:mailto	mailto:info-fax@goto.info.waseda.ac.jp
既存電話	E2U+voice:tel	tel:+81312345678;svc=voice
電話での FAX	E2U+tel	tel:+81312345678;svc=fax
電子メール	E2U+message:mailto	mailto:info@goto.info.waseda.ac.jp
WEB	E2U+web:http	http://www.goto.info.waseda.ac.jp/

第 3 章

ENUM サービスを選択する新しい方法と認証方式

3.1 概要

常時接続環境では、従来の計算機はもちろん、家の中の家電、 AV 機器などのインターネット接続が実現しつつある。これらのマシンに ENUM 機能を付けることで、自分にとって適切な通信手段、通信相手を自動的に選択できるようになる。

一方、たとえ自分が家の中にいなくても、家の中のマシンを操作したいことがときどきある。家に帰る前にエアコンをつけたり、テレビ番組を録画したり、冷蔵庫の内部の品物のチェックなどがある。

そこで、家の外で家の中のネットワーク対応するマシンを操作して、適切な通信を実現することを目指す。図 3.1 は、外にある端末 Controller が家庭内の端末 Sender(あるいは Receiver) にコマンドを送って、別の端末 Receiver(あるいは Sender) と接続させる様子を示している。この Controller はテンキーさえ持っていれば良いので、普及している携帯電話が十分に使える。

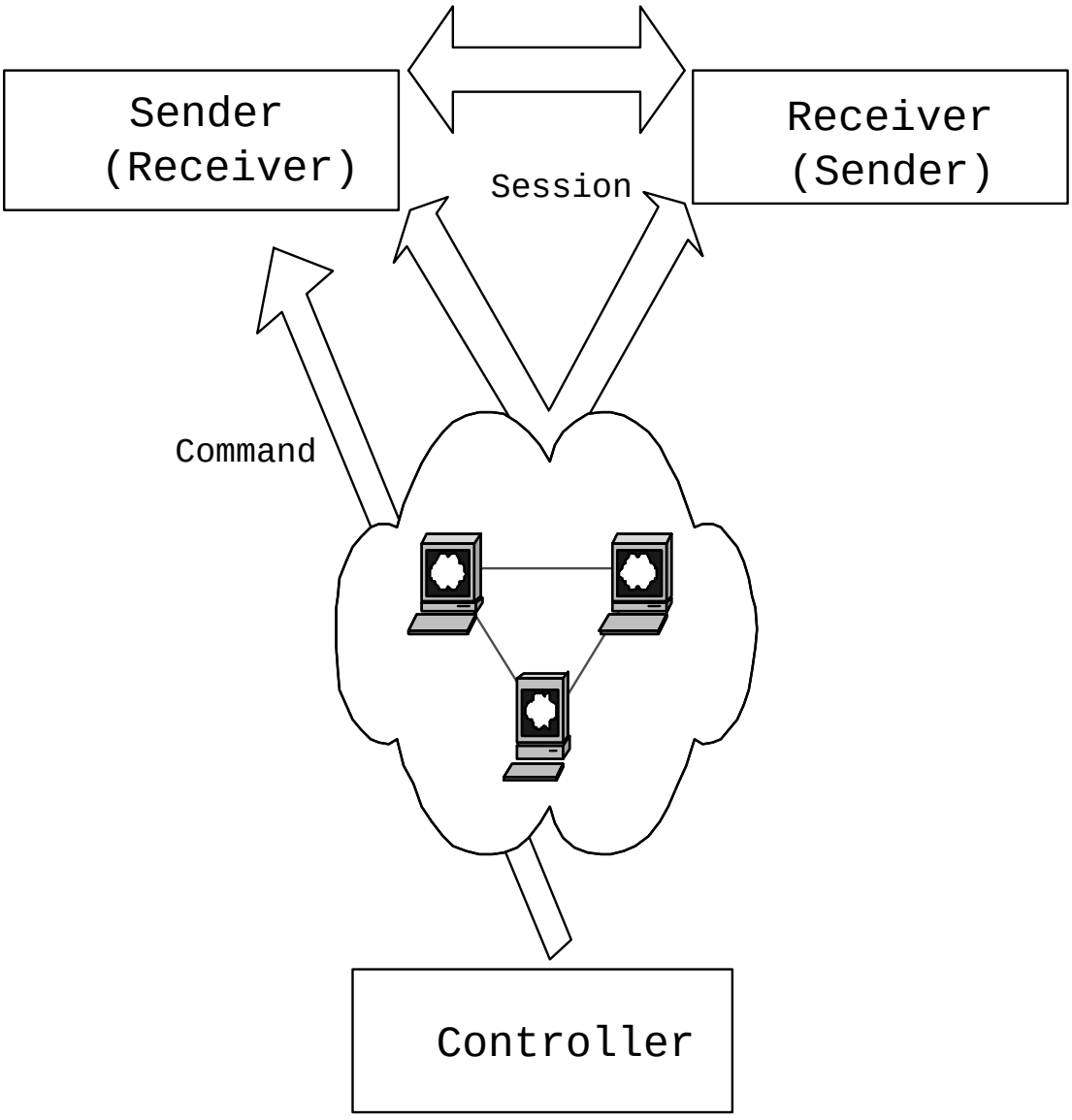
Sender
(Receiver)
Receiver
(Sender)
Session
Command
Controller

図 3.1: 通信の略図

3.2 通信方式

3.2.1 コマンドを送受信する方法

- コマンドを受信する端末は受信するためにポートを 1 つ開けておく。
- Controller 側でコマンドを受信する端末の IP アドレス (あるいはドメイン名) とポート番号を入力して、受信する端末と接続する。
- Controller はコマンドを送信する。

3.2.2 端末の識別方法

端末ごとに十進二桁の番号をつけ、それを識別子とする。 ENUM は番号のみを使って通信できる仕組みなので、アルファベットを含むような名前は使わない。

3.2.3 コマンドの形式

Controller が出すコマンドは次の書式にしたがう。オプションのフィールドは拡張のために用意した。

表 3.1: コマンドの形式

送信端末番号	受信端末番号	オプション
10 進 2 桁	10 進 2 桁	10 進 1 桁

ここでコマンドを受ける端末が自分は Sender かあるいは Receiver かを知るために、オプションのフィールドに ”0” と ”1” を使う。 ”0” は、コマンドを受信する端末が Receiver である。 ”1” は、コマンドを受信する端末が Sender である。

例えば ”97531” は、コマンドをうける端末は端末 97 であり、端末 97 から端末 53 にデータを送信せよ、という意味になる。

3.2.4 AUS

ENUM の AUS は E.164 番号であるが、本研究では AUS として E.164 番号の代わりに、端末につけた番号を使う。そのため、国番号は付かないし、 ”+” も付かない。書式は次のようなる。

問い合わせコードは端末が提供するどの機能を問い合わせるかを区別するために使う。次の表の 3 種類を定義する。

表 3.2: 本研究用 AUS の形式

端末番号	問い合わせコード
10 進 2 桁	10 進 1 桁

表 3.3: 本研究用 AUS の問い合わせコード

コード	問い合わせ内容
0	コマンドの問い合わせ
1	送信機能の問い合わせ
2	受信機能の問い合わせ

コード ”0”は、 Controller が家の中にある場合に使われている。アドレスとポート番号を入力しないで、直接にコマンドを送信するときに、このコードを問い合わせることで、受信する端末の IP アドレスとポート番号を得る。本研究では、 Controller が家の外にあるので、従来使われている DNS を使って、コマンドを受信する端末と接続する。

AUS の例と、その意味を述べる。

340 端末 34 にコマンドを送るための情報を問い合わせるための AUS。

341 端末 34 が持っている送信機能を問い合わせるための AUS。

342 端末 34 が持っている受信機能を問い合わせるための AUS。

3.2.5 key の生成

AUS から ENUM DNS に通信相手の情報を問い合わせるための鍵を生成する。ここでは ”341”という AUS を例に、 AUS から鍵を生成する手順を説明する。

1. 数字の間にドット (”.”) を挿入する。

   3.4.1

2. 数字の並びを逆順にする。

   1.4.3

3. 末尾に ”.mydomain.com.”を付加する。

   1.4.3.mydomain.com.

3.3 認証

端末 Controller が、家の外からインターネットを介して家庭内のマシンをコントロールするため、セキュリティを考える必要がある。つまりコマンドを送るのは誰かを認証する必要がある。そうしないと、誰も簡単に他人の家庭内のネットワーク機器が操作できる。

ここで、コマンドを送る人を認証するためには、 Controller 側でパスワードを暗号化し、コントロールしたいマシンに送信し、受信するマシン側でパスワードを復号化し、保存したコマンドを送信する権利がある人のパスワードと比較し、送信する人が権利があるかどうかを認証する方式を使用する。

世の中で使われている暗号方式は大きく 2 つに分ける。

- 共通鍵暗号
  メッセージの送信者と受信者が同じ鍵を共有しており、それを用いて平文を暗号化、復号化する方法。

- 公開鍵暗号
  対になる 2 つの鍵を使ってデータの暗号化・復号化を行なう暗号方式。
  片方は他人に広く公開するため公開鍵と呼ばれ、もう片方は本人だけがわかるように厳重に管理されるため秘密鍵と呼ばれる。
  秘密鍵で暗号化されたデータは対応する公開鍵でしか復号できず、公開鍵で暗号化されたデータは対応する秘密鍵でしか復号できない。

家庭内のマシンを操作するのは同じ家庭の一員であるので、ここでは共通鍵を使う。本研究では DES 暗号モジュールを Perl に実装し、 8 バイトの鍵と 8 バイトのパスワードを使用する。

3.4 通信の過程

ここまでの検討をもとにして、次のような通信の過程を考える。

1. Controller 側にコマンドを受信するマシンのドメイン名とポート番号を入力し、 DNS サーバに IP アドレスを問い合わせる。
2. DNS サーバから IP アドレスを得る。
3. IP アドレスとポート番号にしたがって、コマンドを受信するマシンと接続する。
4. 受信したマシンが応答し、 Controller にパスワードを要求する。

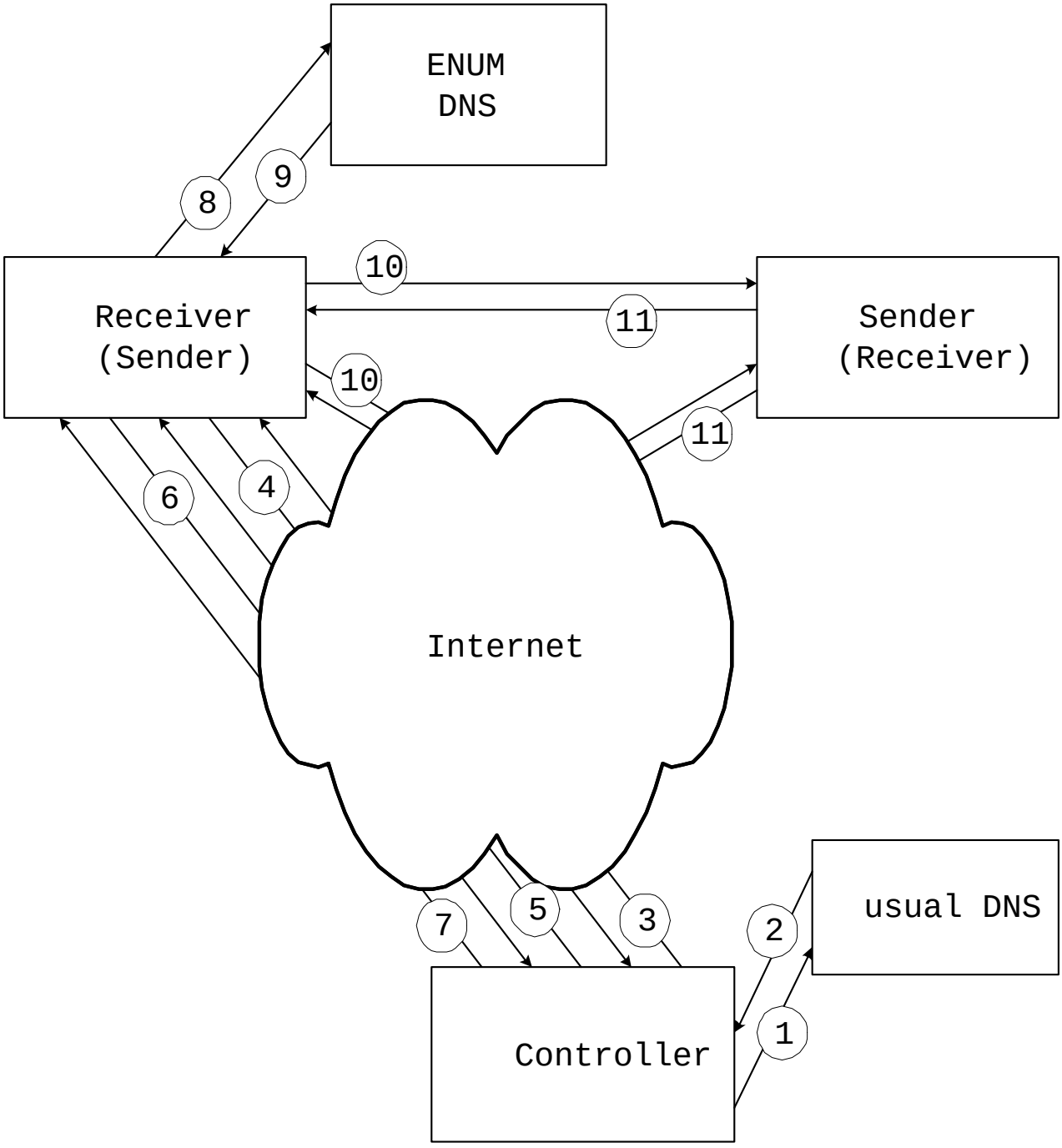
ENUM
DNS
8
9
Receiver
(Sender)
10
11
Sender
(Receiver)
10
11
6
4
Internet
7
5
3
2
usual DNS
Controller
1

図 3.2: 通信の過程

5. Controller 側にパスワードを入力して、共通鍵暗号で暗号化して、相手に暗号化したパスワードを送信する。

6. 暗号文を受信したマシンが暗号文を復号化して、予め保存してあるファイルからパスワードを取り出して、両者を比較する。
   パスワードが正しい場合、 Controller にコマンドを要求する。
   パスワードが正しくない場合、ステップ 4 に戻る。ただし、 3 回連続して正しくない場合、接続を中止する。

7. Controller 側にコマンドを入力し、送信する。コマンドを受信したマシンの応答が返る場合、接続を終了する。

8. コマンドを受信したマシンはコマンドを見て、 Sender の番号、 Receiver の番号、そして自分が Sender であるか Receiver であるかを知る。番号から AUS を生成し、 AUS からデータベースをひく鍵を生成し、鍵をもとに ENUM DNS に次の通信相手の情報を問い合わせる。

9. ENUM DNS サーバからサービス種類と対応する URI が戻ってくる。マシンが適切なサービス種類を選択して、対応する URI を得る。

10. 得た URI をもとに、相手に通信を要求する。

11. 相手が要求を応答し、接続が成立し、通信が始まる。

以上のようにして家の外にある端末 Controller からコマンドを出し、家庭内の端末を操作し、他の端末と通信させる。

この流れを図 3.3 に示す。この図からも分かるように、コマンドを受信する端末の相手は要求を受けてデータを送受信するだけである。本研究では、端末 Controller とコマンドを受信する端末に機能を実装する。

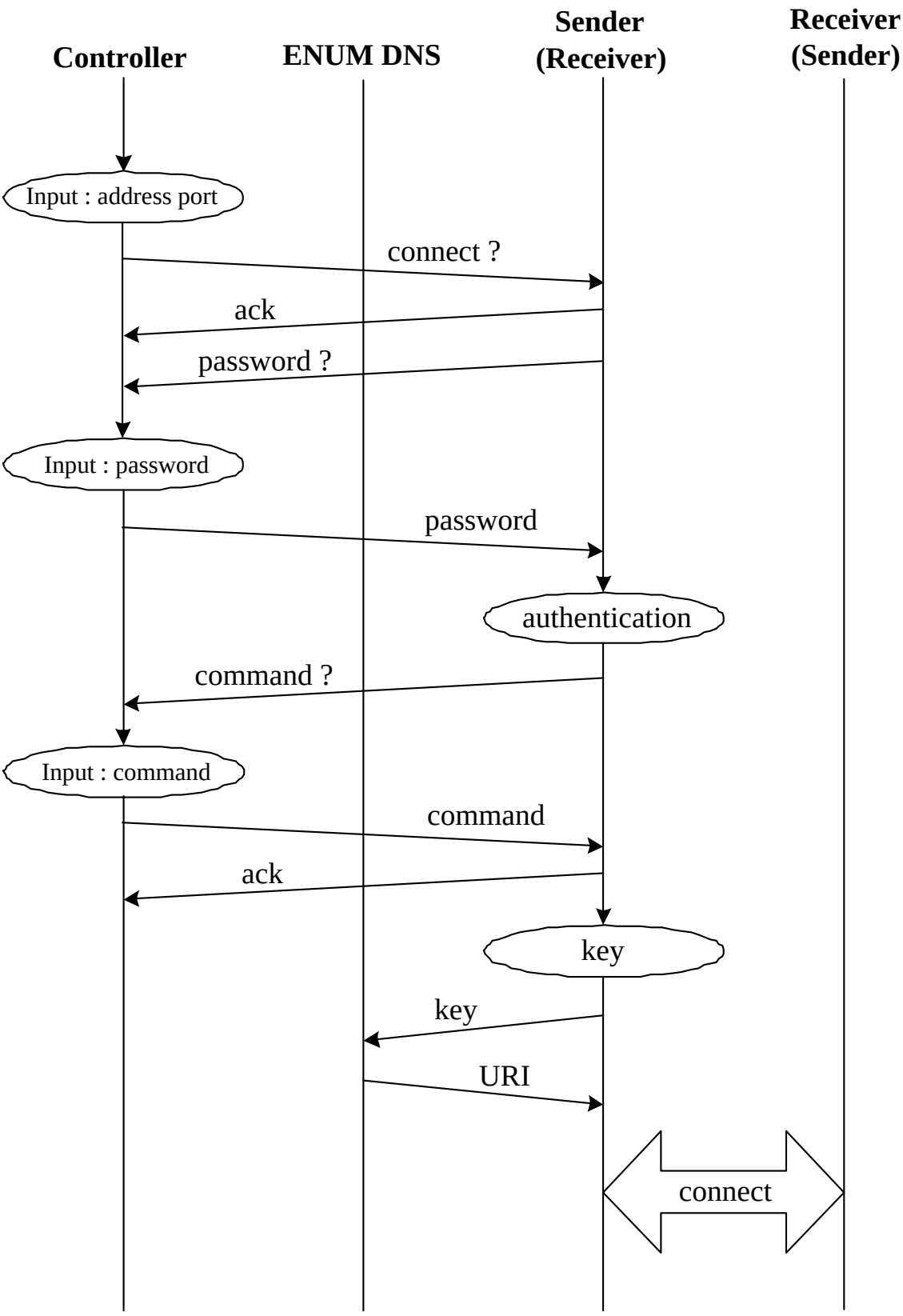
Controller
ENUM DNS
Sender
(Receiver)
Receiver
(Sender)
Input : address port
connect ?
ack
password ?
Input : password
password
authentication
command ?
Input : command
command
ack
key
key
URI
connect

図 3.3: 通信の流れ

第 4 章

実験

ここで実際にプログラムを作成して、第 3 章で提案した方法を検証する。

4.1 実験の環境

端末 Controller は、オペレーティングシステム FreeBSD 4.8-STABLE に、 Perl5 を用いて実装した。コマンドを受信する端末は、オペレーティングシステム Windows XP Professional に、ActivePerl5.8 を用いて実装した。 ENUM DNS サーバはオペレーティングシステム Mac OS X 10.2.8 に、 DNS BIND9.3.0 をインストールした。使用したマシンを表 4.1 に示す。

表 4.1: 使用した PC の仕様

PC	CPU	Memory	OS
Controller	Pentium 200MHz	48MByte	FreeBSD 4.8-STABLE
Receiver	Pentium 4 2.26GHz	1 GByte	Windows XP Professional
ENUM DNS	PowerPC G4 400MHz	128MByte	Mac OS X 10.2.8

プログラムの中で使用したツールを以下に示す。

dig DNS にクエリを出すためのツール

wget Web のデータを一括取得するためのツール

実験に用いたネットワーク構成を図 4.1 に示す。

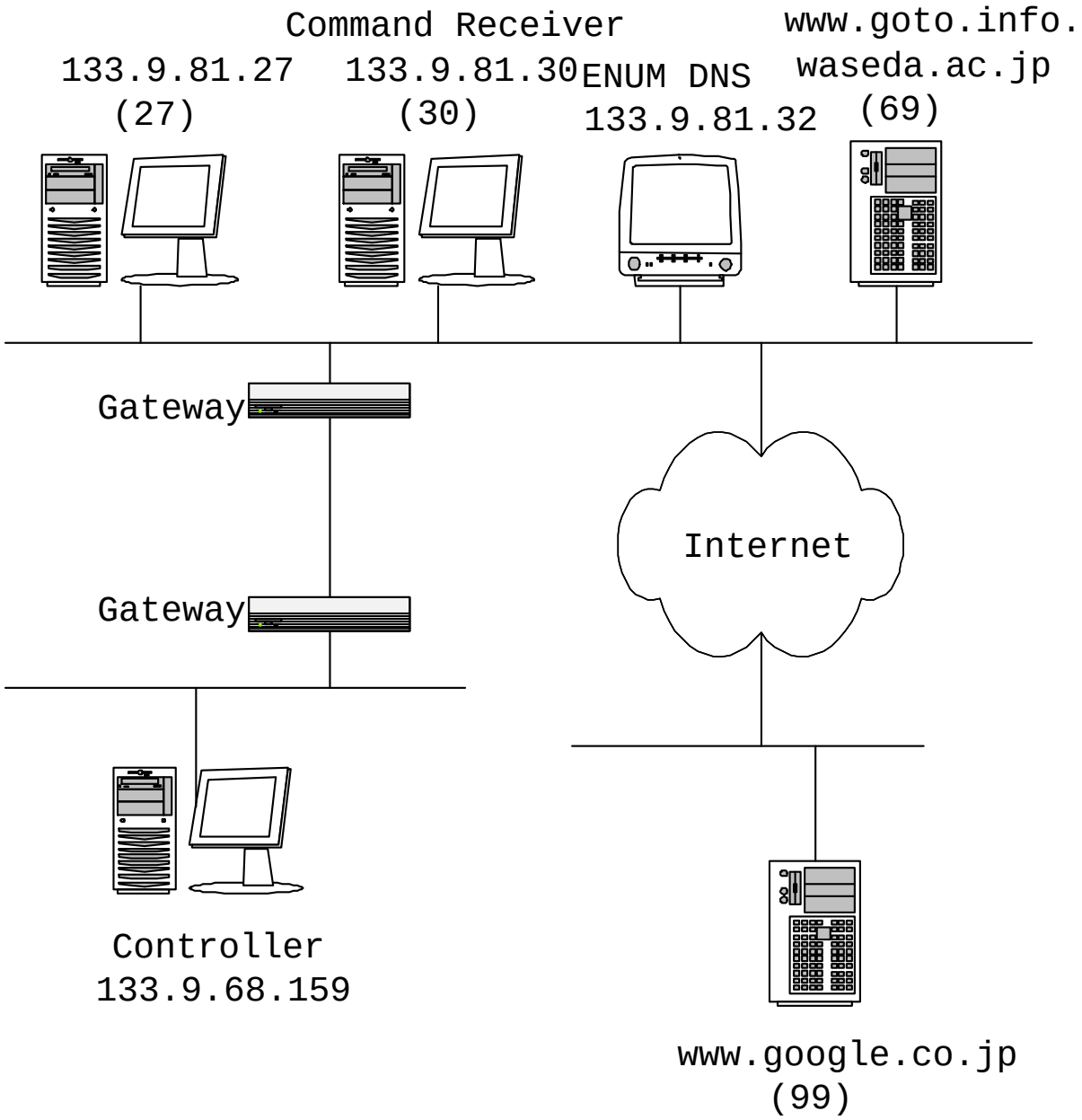

図 4.1: ネットワーク構成図

4.2 実験の内容

コマンドを受信する端末が Sender であるか Receiver であるか、通信相手が同じネットワーク内かネットワークの外かを変えて、実験を行う。

4.2.1 実験 1

コマンドを受信する端末 Sender の端末番号を ”30” とする。通信相手 Recriver は Sender と同じネットワーク内の端末 133.9.81.27 で、端末番号を ”27” とする。端末 Controller から、 ”30271” というコマンドを送信する。

4.2.2 実験 2

コマンドを受信する端末 Receiver の端末番号を ”30” とする。通信相手 Sender は Receiver と同じネットワーク内の端末 www.goto.info.waseda.ac.jp で、端末番号を ”69” とする。端末 Controller から、 ”69300” というコマンドを送信する。

4.2.3 実験 3

コマンドを受信する端末 Receiver の端末番号を ”30” とする。通信相手 Sender はネットワークの外の端末 www.google.co.jp で、端末番号を ”99” とする。端末 Controller から、 ”99300” というコマンドを送信する。

4.3 ENUM DNS サーバの設定

本実験では ENUM DNS サーバには以下のように設定されている。

4.3.1 named.conf

DNS サーバの設定ファイルである named.conf に次の設定を追加する。

```
zone "mydomain.com" {
        type master;
        file "db.mydomain.com";
};
```

4.3.2 ゾーンファイル (db.mydomain.com)

ゾーンファイル (db.mydomain.com) を追加する。

```
$TTL    86400
@       IN      SOA     ns1.mydomain.com.  root.mydomain.com.  (
                                2004011801 ; YYYYMMDDnn
                                10800      ; Refresh
                                3300       ; Retry
                                3600000    ; Expire
                                3600 )     ; Minimum
                IN      NS      ns1.mydomain.com.
               ; IN     NS      ns1.mydomain.com.

ns1         IN      A           133.9.81.32
0.7.2       IN      NAPTR  100 10 "u" "E2U+ft:ftp"
"!^.*$!ftp://133.9.81.27!" .
1.9.6       IN      NAPTR  100 10 "u" "E2U+web:http"
"!^.*$!http://www.goto.info.waseda.ac.jp/!" .
1.9.9       IN      NAPTR  100 10 "u" "E2U+web:http"
"!^.*$!http://www.google.co.jp/!" .
```

4.4 実験の結果

4.4.1 実験 1 の結果

まず Controller の動作を見る。

```
> perl control.pl 133.9.81.30 25739
ip   = 133.9.81.30
port = 25739
Connect succesfully!
```

Controller 端末において control プログラムを起動する。コマンドを受信する端末で accept プログラムを起動した場合、接続が成功する。

```
Do you want to use debug mode?  (y/n) ...  y
```

debug mode というのは、途中のプログラムがどのように動作しているのかを見ることができるモードとして用意してある。

```
input your password:  23456789
password = 23456789
encryptext = ff1856e046c40fb8
Password isn't OK.

input your password:  12345678
password = 12345678
encryptext = 80a00aae2513e193
Password is OK.
```

入力されるパスワードがコマンドを受信する端末に送信されて、保存したパスワードと比較される。正しくない場合、パスワードの再入力が要求される。正しい場合、コマンドを送信するステップに進む。

```
input command:30271
command = 30271
send command
received ack
```

コマンドが入力されて、送信される。 accept プログラムの応答が来る場合には control プログラムが終了する。

次にコマンドを受信する端末の動作を見る。

```
> perl accept.pl
Do you want to use debug mode?  (y/n) ...  y
Waiting connect.
```

端末 Controller に control プログラムを起動する前に、 accept プログラムを debug mode で起動しておいて、端末 Controller からの接続を待っている。

```
Waiting password.
received data = ff1856e046c40fb8
It is under decryption now.
received password = 23456789

Opening the file of the saved password.
saved password = 12345678
Password isn't correct.

received data = 80a00aae2513e193
It is under decryption now.
received password = 12345678

Opening the file of the saved password.
saved password = 12345678
Password is correct.
```

端末 Controller から接続して、パスワードを送信する。パスワードを受信した端末が予め保存しておいたパスワードを取り出して、受信したパスワードと比較する。正しくない場合には、端末 Controller にパスワードの再送信を要求する。正しい場合、コマンドの受信を待つステップに進む。

```
Waiting command.
received command = 30271
Sender   = 30
Receiver = 27
This is Sender.

press 'y' and see AUS and key ...y
AUS = 270
key = 0.2.7.mydomain.com
```

端末 Controller からコマンドを受信して、 Sender は端末 30 で、 Receiver は端末 27、そして自分が Sender であるを知った。通信相手は端末 Receiver(27) であるから、 AUS を 270 と生成する。そして key を生成する。

```
now query for NAPTR.
Do you want to see full reply?  (y/n) ...  y

; <<>> DiG 9.2.3 <<>> @133.9.81.32 0.7.2.mydomain.com naptr
;; global options:  printcmd
;; got answer:
;; ->>HEADER<<- opcode:  QUERY, status:  NOERROR, id:  41
;; flags:  qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;0.7.2.mydomain.com.          IN    NAPTR

;; ANSWER SECTION:
0.7.2.mydomain.com.   86400   IN    NAPTR   100 10 "u" "E2U+ft:ftp"
"!^.*$!ftp://133.9.81.27!" .

;; AUTHORITY SECTION:
mydomain.com.         86400   IN    NS      ns1.mydomain.com.

;; ADDITIONAL SECTION:
ns1.mydomain.com.     86400   IN    A       133.9.81.32

;;Query time:  0 msec
;;SERVER: 133.9.81.32#53(133.9.81.32)
;;WHEN: Tue Dec 28 20:12:47 2004
;;MSG SIZE  rcvd:  125
```

生成した鍵を使い、DNS に問い合わせを出している。プログラム accept では、問い合わせを出すためにツール dig を使っている。ここでは、dig の結果をそのまま表示している。

```
Do you want to see only NAPTR-RR? (y/n) ...  y
0.7.2.mydomain.com.   86400   IN    NAPTR   100 10 "u" "E2U+ft:ftp"
"!^.*$!ftp://133.9.81.27!" .

select and get its regexp
order  ==>100
regexp ==> !^.*$!ftp://133.9.81.27!
connect FTP server.
```

dig の結果のうち、通信相手の情報 (ANSWER SECTION 部分) だけを表示している。得られた結果から、 service フィールドが ”E2U+ft:ftp”で、 IP アドレスが 133.9.81.27 であると知る。そしてこの IP アドレスである ftp サーバと接続する。

4.4.2 実験 2 の結果

端末 Controller の動作は、コマンドを送信する部分を除いて、他の部分は実験 1 と同じである。

```
input command:69300
command = 69300
send command
received ack
```

コマンドを変えて送信する。 accept プログラムの応答が来る場合には control プログラムが終了する。

次にコマンドを受信する端末の動作を見る。

```
Waiting command.
received command = 69300
Sender   = 69
Receiver = 30
This is Receiver.

press 'y' and see AUS and key ...y
AUS = 691
key = 1.9.6.mydomain.com
```

端末 Controller からコマンドを受信して、 Sender は端末 69 で、 Receiver は端末 30、そして

自分が Receiver であることを知った。通信相手は端末 Sender(69) であるから、 AUS を 691 と生成する。そして key を生成する。

```
now query for NAPTR.
Do you want to see full reply?  (y/n) ...  y

; <<>> DiG 9.2.3 <<>> @133.9.81.32 1.9.6.mydomain.com naptr
;; global options:  printcmd
;; got answer:
;; ->>HEADER<<- opcode:  QUERY, status:  NOERROR, id:  41
;; flags:  qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;1.9.6.mydomain.com.          IN    NAPTR

;; ANSWER SECTION:
1.9.6.mydomain.com.   86400   IN    NAPTR   100 10 "u" "E2U+web:http"
"!^.*$!http://www.goto.info.waseda.ac.jp/!" .

;; AUTHORITY SECTION:
mydomain.com.         86400   IN    NS      ns1.mydomain.com.

;; ADDITIONAL SECTION:
ns1.mydomain.com.     86400   IN    A       133.9.81.32

;;Query time:  0 msec
;;SERVER: 133.9.81.32#53(133.9.81.32)
;;WHEN: Tue Dec 29 14:14:07 2004
;;MSG SIZE  rcvd:  144
```

生成した鍵を使い、 DNS に問い合わせを出している。

```
Do you want to see only NAPTR-RR? (y/n) ...  y
1.9.6.mydomain.com.   86400   IN    NAPTR   100 10 "u" "E2U+web:http"
"!^.*$!http://www.goto.info.waseda.ac.jp!" .

select and get its regexp
order  ==>100
regexp ==> !^.*$!http://www.goto.info.waseda.ac.jp!
checking URI ...  OK

press 'y',then execute application ...  y
```

service フィールドが ”E2U+web:http”である。 order と regexp を取り出して、 regexp から URI を生成し、 URI が正しいかどうかを評価する。

評価には wget を使っている。

```
> wget --spider URI
```

を実行すると、 URI で指定されたファイルが存在するかどうか確かめられる。

最後に 'y' を入力すると、ブラウザが起動し、 ”http://www.goto.info.waseda.ac.jp/” が表示された。

4.4.3 実験 3 の結果

端末 Controller の動作は、コマンドを送信する部分を除いて、他の部分は実験 1 と同じである。

```
input command:99300
command = 99300
send command
received ack
```

コマンドを変えて送信する。 accept プログラムの応答が来る場合には control プログラムが終了する。

コマンドを受信する端末の動作は、端末 Sender がネットワークの外であることを除いて、実験 2 と同じである。

```
Waiting command.
received command = 99300
Sender   = 99
Receiver = 30
This is Receiver.

press 'y' and see AUS and key ...y
AUS = 991
key = 1.9.9.mydomain.com
```

端末 Controller からコマンドを受信して、 Sender は端末 99 で、 Receiver は端末 30、そして自分が Receiver であるを知った。通信相手は端末 Sender(99) であるから、 AUS を 991 と生成する。そして key を生成する。

```
Do you want to see only NAPTR-RR? (y/n) ...  y
1.9.6.mydomain.com.   86400   IN    NAPTR   100 10 "u" "E2U+web:http"
"!^.*$!http://www.goto.info.waseda.ac.jp!" .

select and get its regexp
order  ==>100
regexp ==> !^.*$!http://www.goto.info.waseda.ac.jp!
checking URI ...  OK

press 'y',then execute application ...  y
```

ここでは dig の結果のすべては見ずに、 ANSWER SECTION 部分だけ表示させる。 service フィールドが "E2U+web:http"である。 order と regexp を取り出して、 regexp から URI を生成し、 URI が正しいかどうかを評価する。

最後に 'y' を入力すると、ブラウザが起動し、 "http://www.google.co.jp/" が表示された。

第 5 章

結論

5.1 まとめ

本研究の成果を用いれば、家の外から家中の機器を操作し、 ENUM DNS を使って適切なサービスを選択して、通信したい機器と通信できる。こうすることにより、インターネットと接続できる場所、端末であれば、どこでも、いつでも家庭内の機器を制御できる。

本研究では、インターネットを介してコントロールすることを意識して、セキュリティを考慮した。具体的には、認証する方法をプログラムに加えることで、実用化することができた。

5.2 今後の課題

本論文で取扱ったは、 1 つの機器をコントロールし、相手機器と通信させる仕組みである。コントロールしたい機器を識別するために、 1 つの機器に 1 つのグローバル IP アドレスを固定させる方法を使っている。しかし、いまのは IPv4 のアドレスはだんだん少なくなる。一方、家庭内のネットワーク機器の数がどんどん増えている。そんな状況の中で、機器を識別するために、 NAT のようなグローバル IP アドレスからプライベート IP アドレスへの変換システムの取り入れ、あるいは IPv6 への対応が必要になる。

謝辞

本論文の作成にあたり、日頃より多大なる御指導を頂いている後藤滋樹教授に深く感謝致します。また、研究を進める上で貴重なアドバイスを頂いた後藤研究室の史虹波氏、杉田隆俊氏、日頃より助言を頂いた後藤研究室の諸氏に感謝致します。

参考文献

[1] ENUM 研究グループ, 『ENUM 研究グループ報告書』, 社団法人日本ネットワークインフォメーションセンター, 2003.

[2] W. Richard Stevens 著, 橘康夫 訳, 井上尚司 監訳, 『詳解 TCP/IP Vol.1 プロトコル』, ピアソン・エデュケーション, 2000.

[3] P. Faltstrom, RFC2916, 『E.164 number and DNS』, IETF, 2000.

[4] P. Faltstrom & M. Mealling, RFC3761, 『The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)』, IETF, 2004.

[5] ENUM トライアルジャパン (ETJP), 『ENUM トライアルジャパン 第 1 次報告書』, 株式会社日本レジストリサービス, 2004.

[6] ENUM トライアルジャパン (ETJP), 『ENUM トライアルジャパン 第 2 次報告書』, 株式会社日本レジストリサービス, 2004.

[7] Paul Albitz, Cricket Liu 著, 高田広章, 小島育夫 監訳, 『DNS & BIND 第 4 版』, オライリー・ジャパン, 2002.

[8] 杉田隆俊, 『ENUM を応用した三者間の通信法』, 早稲田大学理工学部 2003 年度卒業論文, 2004.

[9] 村山公保, 『基礎からわかる TCP/IP ネットワーク実験プログラミング』, オーム社, 2001.

[10] Larry Wall, Tom Christiansen, Randal L.Schwartz 著, 近藤嘉雪 訳, 『プログラミング Perl 改訂版』, オライリー・ジャパン, 1997.

付録 A

用語の定義

- AUS : Application Unique String
  DDDS アプリケーションへの最初の入力になる文字列。URI にする変換サービスのインプット。

- DDDS : Dynamic Delegation Discovery System
  動的な書き換え規則を反復適用して、文字列を変換する仕組み。

- DNS : Domain Name System
  インターネットに接続された端末の情報を提供する分散データベース。ドメイン名と IP アドレスの対応などをとる。

- ENUM : Telephone Number Mapping
  E.164 番号をキーとして DNS を検索し、その E.164 番号に対応するアプリケーションを URI 形式で得る機構。

- E.164 番号 :
  ITU-T E.164 勧告の付録 A の中で指定された構造、長さ、および唯一性を満たす 10 進の数字列。外国からの着信も可能で、国番号を含めて 15 桁以内の番号。

- E2U : Enum to URI
  DDDS を ENUM で使うプロトコル。

- NAPTR : Naming Authority Pointer
  文字列変換により、ドメインラベルや URI を生成するための書き換え規則を記述するための DNS リソースレコード。

- URI : Uniform Resource Identifiers
  リソースを名前、場所で識別する簡潔に書式化された文字列。

- URL : Uniform Resource Locater
  特定の資源にアクセスする手段 (プロトコル) と資源のある場所を指定する記述様式。

- ゾーン :
  DNS において、あるネームサーバが権威を持つ (管理する)DNS データベースの一部のこと。

- リソースレコード (RR : Resource Record) :
  DNS データベースの各構成要素。 NAPTR も RR の一つ。